

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-225694

(43)Date of publication of application : 22.08.1995

(51)Int.Cl.

G06F 9/46

(21)Application number : 06-015089

(71)Applicant : HITACHI LTD

(22)Date of filing : 09.02.1994

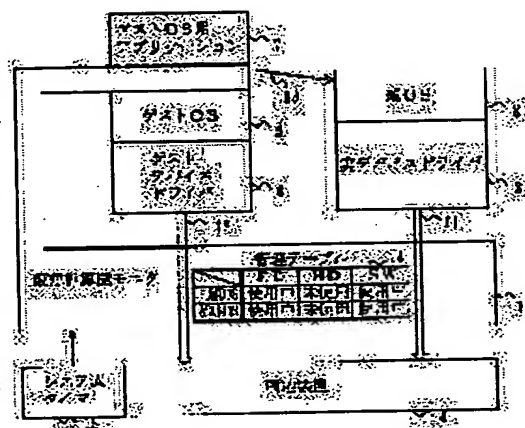
(72)Inventor : YAMADA YOSHIHIRO  
MAKIOKA JUNICHI

## (54) VIRTUAL COMPUTER SYSTEM

### (57)Abstract:

**PURPOSE:** To provide a virtual computer system capable of smoothly performing a processing even when the same peripheral equipment is alternately utilized from plural OSes.

**CONSTITUTION:** A virtual computer 1 is provided with a table 14 for storing the state of the peripheral equipment 4 controlled by a real device controller 3 and the state of the peripheral equipment controlled by a quest device driver 6. Then, by monitoring the respective device drivers 3 and 6 accessing the peripheral equipment 4 by the virtual machine monitor 1, checking the table 14 for storing the states of the peripheral equipment and permitting access only when conditions are valid, the access of the peripheral equipment by the plural OSes is smoothly performed.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

BEST AVAILABLE COPY

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号 ✓

特開平7-225694

(43)公開日 平成7年(1995)8月22日

(51)Int.Cl.<sup>6</sup>

G 0 6 F 9/46

識別記号

3 5 0

庁内整理番号

7737-5B

F I

技術表示箇所

審査請求 未請求 請求項の数 3 O L (全 7 頁)

(21)出願番号 特願平6-15089

(22)出願日 平成6年(1994)2月9日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 山田 善弘

神奈川県横浜市戸塚区吉田町292番地株式

会社日立製作所映像メディア研究所内

(72)発明者 牧岡 淳一

茨城県日立市東多賀町一丁目1番1号株式

会社日立製作所情報映像メディア事業部内

(74)代理人 弁理士 小川 勝男

(54)【発明の名称】 仮想計算機システム

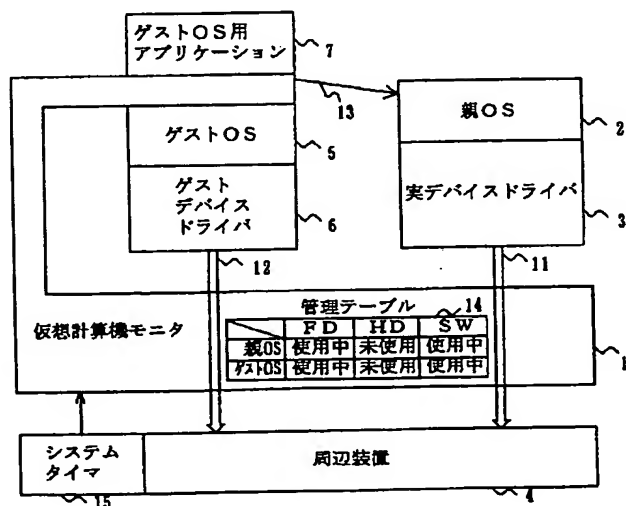
(57)【要約】

【目的】複数のOSから同一の周辺機器を交互に利用しても円滑に処理の行える仮想計算機システムを提供すること。

【構成】仮想計算機モニタに実デバイスコントローラにより制御される周辺装置の状態と、ゲストデバイスドライバにより制御される周辺機器の状態を格納するテーブルを設ける。

【効果】各デバイスドライバが周辺装置をアクセスするのを仮想計算機モニタで監視し、周辺機器の状態を格納するテーブルをチェックして、条件が成立する場合のみアクセスを許可することにより、複数のOSによる周辺機器のアクセスが円滑に行うことができる。

図1



## 【特許請求の範囲】

【請求項1】 実計算機を制御する親OSと、該親OSの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストOSと、ゲストOSで動作するアプリケーションから成り、ゲストOS上のアプリケーションから親OSの機能を直接利用することのできるインターフェイスを有することを特徴とする仮想計算機システム。

【請求項2】 実計算機を制御する親OSと、該親OSの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストOSと、親OSで動作するアプリケーションから成り、親OS上のアプリケーションからゲストOSの機能を直接利用することのできるインターフェイスを有することを特徴とする仮想計算機システム。

【請求項3】 実計算機を制御する親OSと、周辺装置を制御する実デバイスハンドラと、該親OSの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストOSと、ゲストOSから実計算機の周辺装置を制御するゲストデバイスドライバとゲストOSで動作するアプリケーションから成り、前記仮想計算機モニタに実デバイスコントローラにより制御される周辺装置の状態と、ゲストデバイスドライバにより制御される周辺機器の状態を格納するテーブルを設け、前記テーブルの内容を監視して実際の周辺機器を操作する手段を設けたことを特徴とする仮想計算機システム。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、一つの実計算機上で複数の異なるOSを利用する仮想計算機に係わり、特に、ゲストOS上のアプリケーションから親OSのサービスを直接利用する事ができ、さらに、ゲストOSと親OSから効率良く実計算機の周辺機器を制御することのできる仮想計算機システムに関する。

## 【0002】

【従来の技術】 複数のOSを同時に動かすような仮想計算機システムにおいては、実計算機のハードウェアに対して、2つ以上のOSから同時に入出力要求が発生しないようにするのが普通である。例えば特開平3-131936号公報では、仮想計算機モニタに、ゲストOSによる入出力装置へのアクセスを管理するためのテーブルを設けて2つ以上のOSからの入出力装置への要求が同時に起きないように管理している。

【0003】 しかし、従来の方式では、ゲストOS上のアプリケーションが、親OSのサービスによる周辺装置へのアクセスと、ゲストOSのサービスによる周辺装置へのアクセスを交互に用いる必要がある場合には、周辺装置へのアクセスごとに各OSが前記周辺装置を使い始めてから処理を終了して解放するまでに別のOSによ

るアクセスができず、処理を円滑に行えない可能性があった。

## 【0004】

【発明が解決しようとする課題】 本発明の目的は、複数のOSを同時に動かすことができ、ゲストOS上のアプリケーションからゲストOSのサービスと親OSのサービスを交互に用いて一つの周辺装置をアクセスした場合でも円滑に周辺機器の利用が可能な仮想計算機システムを提供することにある。

## 10 【0005】

【課題を解決するための手段】 前記課題を解決するために、本発明では、実計算機を制御する親OSと、該親OSの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストOSと、ゲストOSで動作するアプリケーションから成る仮想計算機システムにおいてゲストOS上のアプリケーションから親OSの機能を直接利用することのできるインターフェイスを設ける。また、実計算機を制御する親OSと、該親OSの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストOSと、親OSで動作するアプリケーションから成る仮想計算機システムにおいて親OS上のアプリケーションからゲストOSの機能を直接利用することのできるインターフェイスを設ける。さらに、実計算機を制御する親OSと、周辺装置を制御する実デバイスハンドラと、該親OSの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストOSと、ゲストOSから実計算機の周辺装置を制御するゲストデバイスドライバとゲストOSで動作するアプリケーションから成る仮想計算機システムにおいて、前記仮想計算機モニタに実デバイスコントローラにより制御される周辺装置の状態と、ゲストデバイスドライバにより制御される周辺機器の状態を格納するテーブルを設け、前記テーブルの内容を監視して実際の周辺機器を操作する手段を設ける。

## 【0006】

【作用】 実計算機を制御する親OSと、該親OSの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストOSと、ゲストOSで動作するアプリケーションから成る仮想計算機システムにおいて、ゲストOS上のアプリケーションから特定の割込を発生させることにより、仮想計算機モニタに制御を移すようにする。仮想計算機モニタは、特定の割込が発生すると、親OSを起動して親OSのサービスを要求してサービスを行わせる。親OSのサービスが終了した後にゲストOSに復帰することにより、ゲストOS上のアプリケーションから親OSの機能を直接利用することができるようにする。

【0007】 また、実計算機を制御する親OSと、該親OSの動作を監視する仮想計算機モニタと、該仮想計算

機モニタの制御のもとで動作する一つ以上のゲストOSと、親OSで動作するアプリケーションから成る仮想計算機システムにおいて、親OS上のアプリケーションから特定の割込を発生させることにより、仮想計算機モニタに制御を移すようにする。仮想計算機モニタは、特定の割込が発生すると、ゲストOSを起動してゲストOSのサービスを要求してサービスを行わせる。ゲストOSのサービスが終了した後に親OSに復帰することにより、親OS上のアプリケーションからゲストOSの機能を直接利用することができるようにする。

【0008】さらに、ゲストOS上のアプリケーションから親OSを通じての周辺機器のアクセスとゲストOSを通じての周辺機器のアクセスを交互に行う場合でも、仮想計算機モニタに実デバイスコントローラにより制御される周辺装置の状態と、ゲストデバイスドライバにより制御される周辺機器の状態を格納するテーブルを設け、各デバイスドライバが周辺装置をアクセスするのを仮想計算機モニタで監視して、どちらかのデバイスドライバが周辺機器にアクセスする時点で、周辺機器の状態を格納するテーブルをチェックして、条件が成立する場合のみ周辺機器に対するアクセスを許可することにより、複数のOSによる周辺機器のアクセスが円滑に行えるようにする。

【0009】

【実施例】図1は、本発明の一実施例の構成図である。図中、構成要素1は仮想計算機モニタ、2は親OS、3は実デバイスドライバ、4は実計算機の周辺装置、5はゲストOS、6はゲストデバイスドライバ、7はゲストOS用アプリケーション、11は実デバイスドライバからのハードウェアアクセス、12はゲストデバイスドライバからのハードウェアへのアクセス、13はゲストOS用アプリケーションからの親OSの呼出、14は仮想計算機モニタ内の周辺機器の状態管理テーブル、15はシステムタイマである。

【0010】図2は、本発明の動作を示すフロー図である。図中、番号200から215は、処理内容を示す。

【0011】仮想計算機モニタ1は、親OS2およびゲストOS5よりも高い特権を持っており、各OSの動作を監視するものとする。立ちあげ時はまず、親OSをはじめに立ちあげる。親OSは特権レベルの高いスーパーバイザモードをサポートしないものとする。親OSを立ちあげてから親OSにより仮想計算機のプログラムを起動する。仮想計算機のプログラム自体がスーパーバイザモードを利用してしまふことにより仮想計算機モニタ1が親OSよりも高い特権レベルを持つことができる。なお、ゲストOSもスーパーバイザモードを用いないと仮定して以後の説明を続ける。仮想計算機モニタ1はCPUからのI/Oアクセスを監視することにより周辺装置へのアクセスを検出する。また、ソフトウェア割り込みを監視することによりアプリケーションからのOSに対

する呼出を検出する。さらに、仮想計算機モニタはハードウェア割り込みを監視することにより周辺機器からの割込を検出する。

【0012】親OS2は周辺装置4を有する実計算機を用いるために設計された本来のOSであり、実デバイスドライバ3というプログラムにより周辺装置4の制御を行う。矢印11は実デバイスドライバによる周辺機器4へのアクセスを示す。アクセス11は前述のように仮想計算機モニタにより監視されている。

10 【0013】ゲストOS5は、親OS2用のハードウェアとは異なるハードウェアを対象に作られたOSであり、ゲストOS用アプリケーション7を動かすことができる。ゲストOS5はゲストデバイスドライバ6により周辺装置4にアクセスする。ここでは、ゲストデバイスドライバ6は周辺装置4を制御可能なように作られているものとする。

【0014】本来、ゲストOS用アプリケーション7を周辺装置4を有する実計算機で動作させるためには、アプリケーション7を親OS用に作り替えることが必要である。しかし、アプリケーション7の規模が大きい場合は、アプリケーション7の変更のための手間が非常に大きなものになってしまう。そこで、周辺装置4を有する実計算機の上で仮想計算機モニタ1を動作させ、仮想計算機モニタ1の管理の下でゲストOS5およびゲストOS5用のアプリケーションを利用することにより、異なるハードウェアの上でもアプリケーション7を利用することを可能とする。

【0015】計算機のハードウェアが異なると、メモリマップが異なる場合が多い。そこで仮想計算機モニタ1は、ゲストOS5およびゲストOS用アプリケーション7が動作するためのメモリマップを実現する機能を有するものとする。

【0016】はじめに、ゲストOS用アプリケーション7がゲストOS5に対してサービスを要求した場合の動作から説明する。ゲストOS用アプリケーション7はゲストOS5に対してソフトウェア割り込みの形でOSのサービスを要求する。OSのサービスとしては例えば外部記憶装置に対する読み書きの要求や、入出力装置に対する入出力の要求などがある。

40 【0017】処理200においてソフトウェアの割込命令が実行されると、処理201において仮想計算機モニタ1が起動される。仮想計算機モニタ1は割込の種類を判断し、それがゲストOS5に対するサービスの要求の場合は、処理202のゲストOS5の割込処理ルーチンに制御を移す。ゲストOS5は処理203のゲストデバイスドライバ6を利用して周辺装置4をアクセスし、サービスを実行する。サービス終了後に、処理205でゲストOS5からゲストOS用アプリケーション7に復帰する。

50 【0018】次に、ゲストOS用アプリケーション7か

5

ら親OS 2のサービスを要求する場合の説明を行う。処理200でゲストOS用アプリケーション7はソフトウェア割り込みの形でOSのサービスを要求する。ソフトウェアの割込命令が実行されると、処理201で仮想計算機モニタ1が起動される。仮想計算機モニタ1は割込の種類を判断する。ここでは、特定の割込番号をあらかじめ決めておき、特定の割込番号の場合は親OS 2のサービスを要求する割込とわかるようにしておく。割込が発生して、それが親OS 2に対するサービスの要求の場合は、仮想計算機モニタ1は処理212で、親OS 2の割込処理ルーチンを実行させるためにまず、親OS 2の動作可能なメモリマップを設定してから親OS 2の割込処理ルーチンに制御を移す。親OS 2は処理213で実デバイスドライバ3を利用して周辺装置4をアクセスし、サービスを実行する。サービス終了後に、仮想計算機モニタ1を呼び出すプログラムを実行して仮想計算機モニタ1に制御を移す。ここでは、また、割込処理から復帰する命令を仮想計算機モニタ1で監視して割込処理の終了を検出しても良い。仮想計算機モニタ1はゲストOS 2が実行可能なメモリマップに設定してからゲストOS用アプリケーション7に復帰する。

【0019】以上の方式で、ゲストOS用アプリケーション7から親OS 2のサービスを受けることが可能となる。この方式の利点は、例えばゲストOS 5で用いるフロッピディスクのファイル構造と親OSで用いるフロッピディスクのファイル構造が異なる場合でも、一つのディスクドライブにゲストOS用のフロッピディスクをいれておきもう一つのフロッピディスクドライブに親OS用のフロッピディスクをいれておき、ゲストOSのアプリケーション7が自由に両方のデータ構造を有するフロッピディスクにアクセスしてデータやファイルの読み書きを行うことを可能とすることができる。

【0020】次に、2個以上の異なるOSとデバイスドライバにより、一つの周辺機器にアクセスする場合の問題点およびその解決手段について説明する。一般的な方法としては、一つのOSが周辺機器にアクセスする場合は、別のOSはその周辺機器に対するアクセスは一切禁止してしまうのが普通であり、これを排他制御と呼ぶ。ところが、例えばフロッピディスクやハードディスクのように媒体をモータで回転させてデータを読み書きする周辺機器では、データの読み書きが終了しても数秒から数分の間モータを回し続けることが多い。これはモータの起動および停止には有る程度の時間を必要とするので、一度フロッピディスクへのアクセスが行われてモータを回転させた場合は、しばらくの間回転させた状態にし、フロッピディスクへのアクセスが一定期間の間になかった場合のみ消費電力の低減やディスクドライブの機械的な摩耗を防ぐためにモータを止めるようにしているシステムが多い。このモータの起動および停止の制御をデバイスドライバによりソフトウェアで行っている場合

6

は問題が起きる可能性が有る。

【0021】本実施例における実計算機システムにおいては、フロッピディスクのモータの制御は、システムタイマー15により行われる。デバイスドライバ3により、フロッピディスクの読み書きを行う場合、デバイスドライバ3はまず、デバイスドライバ3のワークエリア内のカウンタを特定の値に設定してからフロッピディスクのモータを起動状態にする。モータの起動にはある程度の時間がかかるので時間待ちを行ってからフロッピディスクの読み書きを行う。

【0022】次にフロッピディスクの読み書きを行う場合には、モータの状態を検出して、モータがオフ状態の場合はカウンタを設定してからモータを起動して読み書きを行う。また、モータが回転している状態の場合はカウンタを一定の値に設定してすぐに読み書きを行うことができる。

【0023】モータの停止はシステムタイマー割り込みにより行う。実計算機の周辺装置4にはシステムタイマー15が設けられており、1秒間に数十回の割合でシステムタイマー割り込みが発生する。実計算機の場合は、システムタイマー割り込みが発生すると、親OSのシステムタイマー割込処理ルーチンに制御が移されシステムタイマーの処理が行われる。システムタイマー割込でフロッピディスクのモータ制御を行う場合は、システムタイマー割り込み処理ルーチンでまず、モータ制御用のカウンタの値を1だけ減らす。もしも、0になった場合は、モータを停止するようにする。ここで、システムタイマーが1秒に20回発生する場合、フロッピディスクのモータを10秒間アクセスが無かった場合に停止したい場合は、デバイスドライバによりカウンタの初期値を200に設定すれば良い。

【0024】ゲストOS 5によりフロッピディスクにアクセスする場合も親OSとほぼ同じシーケンスで制御を行う。

【0025】ここで、一つのディスクドライブに、親OS 2のファイル構造のフロッピディスクをセットして、もう一つのディスクドライブにゲストOSのファイル構造を有するフロッピディスクをセットし、ゲストOS用のアプリケーション7から交互にゲストOS 5と親OS 2のサービスを利用して両方のフロッピディスクにアクセスする場合を考える。ここで、ゲストデバイスドライバ6と実デバイスドライバ3はそれぞれの内部にモータ制御用のカウンタを用意しており、システムタイマー15からの割込がかかると、仮想計算機モニタによりゲストOS 5側の割込処理ルーチンおよび親OS 2側の割込処理ルーチンの両方の割込処理ルーチンによる処理を行うものとする。また、ディスクドライブのモータはどのドライブも同時に起動および停止の制御を受けるものとする。

【0026】ゲストOS用のアプリケーション7が親OS

S2により親OS2のファイル構造を有するフロッピディスクの読み書きを行った後で、ゲストOS5により別のディスクドライブにセットされたゲストOS5のファイル構造を有するフロッピディスクをアクセスしている場合、親OS2によりフロッピディスクをアクセスしてからしばらくすると、システムタイマ15による割込処理ルーチンにより、ゲストOS5でフロッピディスク使用中にもかかわらず強制的にモータを停止してしまう可能性がある。

【0027】前述の問題は、仮想計算機モニタに両方のOSによるモータの状態を保持するテーブルを設けて、どちらかのOSのデバイスドライバがモータを停止しようとした場合に、前記のテーブルの内容をチェックして、どのOSのデバイスドライバにおいてもモータが停止になる場合のみ実際にモータの停止を行うようにすることで解決することができる。

【0028】仮想計算機モニタ1は内部に管理テーブル14を持ち、さらに、実デバイスドライバ3からの周辺装置4へのアクセス11およびゲストデバイスドライバ6からの周辺装置4へのアクセス12を監視する。これは、CPUが特定のI/Oポートへのアクセスを検出する機能を利用することにより実現する。

【0029】フロッピディスクのモータを制御するI/Oポートへのアクセスが有った場合に、仮想計算機モニタ1呼び出されるようにしておき、モータ起動時は、管理テーブル14にどのOSでモータを起動状態にしたかという情報を設定する。どのOSからI/Oポートがアクセスされたのかを知るためには、現在動作中のOSを識別するためのフラグを設けて、仮想86モニタによりそのフラグを管理する方法がある。また、I/Oポートアクセス時に実行されていたプログラムの置かれているアドレスを調べることにより、どのOSが実行中かを識別することも可能である。

【0030】システムタイマ15によるタイマ割込により、仮想計算機モニタ1はゲストOS5の割込処理ルーチンと親OS2の割込処理ルーチンの両方の処理を行わせる。モータの停止は、システムタイマ15がタイマ割込を発生することにより割込処理ルーチンがカウンタをへらし、カウンタの値が0になった場合に行われる。割込処理ルーチンがI/Oポートをアクセスしてモータを停止しようとした場合に仮想計算機モニタはそれを検出して管理テーブル14に、どのOSでモータが停止状態になるかを設定する。次に、他のOSによるモータの状況を調べて、全てのOSにおいてモータの状態が停止状態になることを確認した場合のみ実際にI/Oポートをアクセスしてモータを停止状態にする。なお、他のOSで一つでもモータが回転状態になっている場合は、テーブル14を設定するだけで実際にはI/Oポートへのアクセスは行わないようにする。

【0031】本発明は外部記憶装置以外にも、リレーに

より計算機自身が計算機の電源を遮断するようなシステムにおいても応用できる。例えば親OS2が動作中にゲストOS5から電源が遮断されるとシステムに障害が発生する可能性があるが、仮想計算機モニタ1において、各OSの電源遮断状況を持つフラグを用意して、全てのOSにおいて電源が遮断されたときのみ実計算機の電源を遮断することにより障害の発生を回避することが可能となる。

【0032】図3は、本発明の一実施例の構成図である。図中、構成要素1～15は図1の構成要素と同様の要素を示すので説明は省略する。307は親OS用アプリケーション、313は親OS用アプリケーションからのゲストOSの呼出である。

【0033】通常、親OSのアプリケーションがゲストOSのサービスを必要とすることはあまり無いが、親OSの開発環境がゲストOSのそれよりも優れている場合は、親OS上でゲストOSのアプリケーションのサポート用ソフトウェアを開発することが有り得る。例えば、ゲストOS用のフロッピディスクの読み書きが必要な場合、親OS上でうごくアプリケーションからゲストOSのサービスを利用したい場合が出てくる。

【0034】その場合、親OSのアプリケーションから、ゲストOSのサービスを要求する手段が必要となる。次に、親OS用アプリケーション307がゲストOS5のサービスを要求する場合の説明を行う。親OS用アプリケーション307はソフトウェア割り込みの形でOSのサービスを要求する。ソフトウェアの割込命令が実行されると、仮想計算機モニタ1が起動される。仮想計算機モニタ1は割込の種類を判断する。ここでは、特定の割込番号をあらかじめ決めておき、特定の割込番号の場合はゲストOS5のサービスを要求する割込とわかるようにしておく。割込が発生して、それがゲストOS5に対するサービスの要求の場合は、仮想計算機モニタ1は、ゲストOS5の割込処理ルーチンを実行させるためにまず、ゲストOS5の動作可能なメモリマップを設定してからゲストOS5の割込処理ルーチンに制御を移す。ゲストOS5はゲストデバイスドライバ6を利用して周辺装置4をアクセスし、サービスを実行する。サービス終了後に、仮想計算機モニタ1を呼び出すプログラムを実行して仮想計算機モニタ1に制御を移す。ここでは、また、割込処理から復帰する命令を仮想計算機モニタ1で監視して割込処理の終了を検出しても良い。仮想計算機モニタ1は親OS2が実行可能なメモリマップに設定してから親OS用アプリケーション307に復帰する。

【0035】以上の方式で、親OS用アプリケーション307からゲストOS5のサービスを受けることが可能となる。この方式の利点は、例えば親OS2で用いるフロッピディスクのファイル構造とゲストOS5で用いるフロッピディスクのファイル構造が異なる場合でも、一

つのディスクドライブにゲストOS用のフロッピディスクをいれておきもう一つのフロッピディスクドライブに親OS用のフロッピディスクをいれておき、親OSのアプリケーション2が自由に両方のデータ構造を有するフロッピディスクにアクセスしてデータやファイルの読み書きを行うことを可能とすることなどに有る。

#### 【0036】

【発明の効果】本発明によれば、実計算機を制御する親OSと、該親OSの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストOSと、ゲストOSで動作するアプリケーションから成る仮想計算機システムにおいて、ゲストOS上のアプリケーションから特定の割込を発生させることにより、仮想計算機モニタに制御を移すようにし、親OSを起動して親OSのサービスを行わせる。親OSのサービスが終了した後にゲストOSに復帰することにより、ゲストOS上のアプリケーションから親OSの機能を直接利用することができるようにする。

【0037】また、実計算機を制御する親OSと、該親OSの動作を監視する仮想計算機モニタと、該仮想計算機モニタの制御のもとで動作する一つ以上のゲストOSと、親OSで動作するアプリケーションから成る仮想計算機システムにおいて、親OS上のアプリケーションから特定の割込を発生させることにより、仮想計算機モニタに制御を移すようにし、ゲストOSを起動してゲストOSのサービスを行わせる。ゲストOSのサービスが終了した後に親OSに復帰することにより、親OS上のアプリケーションからゲストOSの機能を直接利用することができるようにする。

【0038】さらに、ゲストOS上のアプリケーションから親OSを通じての周辺機器のアクセスとゲストOS

を通じての周辺機器のアクセスを交互に行う場合でも、仮想計算機モニタに実デバイスコントローラにより制御される周辺装置の状態と、ゲストデバイスドライバにより制御される周辺機器の状態を格納するテーブルを設け、各デバイスドライバが周辺装置をアクセスするのを仮想計算機モニタで監視して、どちらかのデバイスドライバが周辺機器にアクセスする時点で、周辺機器の状態を格納するテーブルをチェックして、条件が成立する場合のみ周辺機器に対するアクセスを許可することにより、複数のOSによる周辺機器のアクセスが円滑に行うことができる。

#### 【図面の簡単な説明】

【図1】本発明の1実施例を示す構成図である。

【図2】処理内容の説明図である。

【図3】本発明の1実施例を示す構成図である。

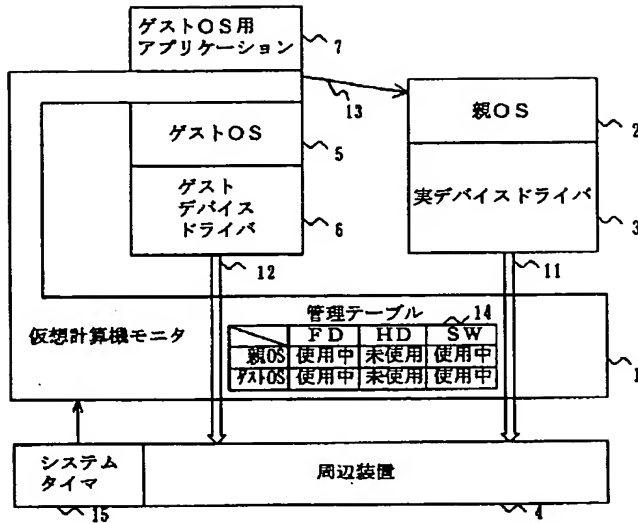
#### 【符号の説明】

- 1…仮想計算機モニタ、
- 2…親OS、
- 3…実デバイスドライバ、
- 4…周辺装置、
- 5…ゲストOS、
- 6…ゲストデバイスドライバ、
- 7…ゲストOS用アプリケーション、
- 11…実デバイスドライバによるアクセス、
- 12…ゲストデバイスドライバによるアクセス、
- 13…親OSへのインターフェイス、
- 14…管理テーブル、
- 200～215…処理内容、
- 307…親OS用アプリケーション、
- 313…ゲストOSへのインターフェイス。



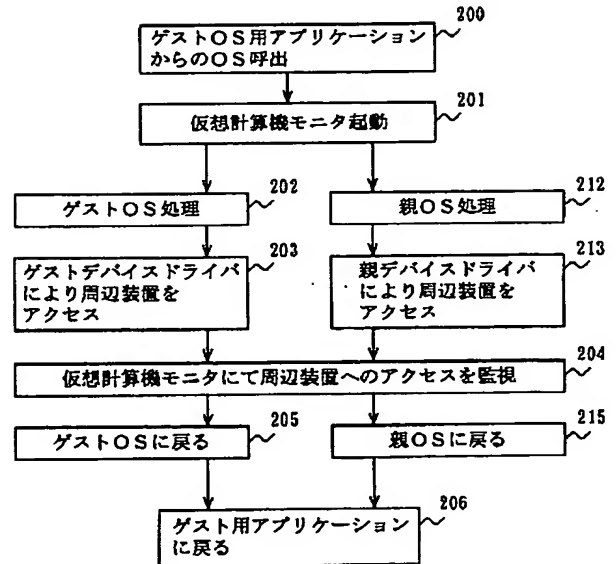
【図 1】

図 1



【図 2】

図 2



【図 3】

図 3

